Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 32 (2014) 1071 – 1078

Procedia Computer Science

International Workshop on Recent Advances on Machine-to-Machine Communication (RAMCOM 2014)

Timed Verification of Machine-to-Machine communications

Ghada Gharbi[c,a,*], Nawal Guermouche[b,a], Thierry Monteil[b,a]

[a] CNRS, LAAS, 7, avenue du colonel Roche, F-31400 Toulouse, France
[b] Université de Toulouse, INSA, F-31400
[c] Université de Toulouse, UPS, F-31400

Abstract

Machine-to-Machine communications (M2M) is emerging as a new technology for next generation of communicating systems. In such M2M based systems, a large number of intelligent machines share information and make collaborative decisions without human interventions. M2M is involved in several fields and in particular, in critical ones such as the healthcare services, where requirements in term of Quality of Services (QoS) and particularly time related requirements must be thoroughly enforced. In this paper, we are interested in the verification problem of Publish/ Subscribe based M2M systems where we consider time related requirements. In this context, we first characterize Publish/ Subscribe communications on the top of the OM2M platform, which is an implementation of the ETSI M2M standard. Then, we introduce a formal graph based model to specify Publish/ Subscribe communications associated to time related requirements. Based on the defined formal model, we provide a model checking based techniques to validate time requirements for data exchange delays.

Keywords: Machine-to-Machine, ETSI M2M architecture, Publish/ Subscribe communication pattern, temporal verification, graph-based model, model checking techniques, OM2M

1. Introduction

Machine-to-Machine (M2M) paradigm is defined as a new technology for next generation of communication where large number of intelligent machines share information and collaborate to make decisions without the human intervention. To enable communication between M2M devices and applications, several communication models are used such as event-oriented communications. Publish/ Subscribe pattern is a promising alternative to ensure event-based communications. The strength of this event-based interaction style lies in the full decoupling in time, space, and the selective data delivery between events producers and consumers which enforces scalability, one issue of M2M systems. M2M is involved in a wide range of applications, for example: automotive, smart metering, e-Health and smart grids. These systems share common characteristics and requirements such as network heterogeneity, mobility and timed sensibility of data transmission [1]. Ensuring Quality-of-Service (QoS), more specifically timed related QoS

* Corresponding author. Tel.: +33 6-58-97-18-25.
E-mail address: gharbi@laas.fr

1877-0509 © 2014 Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and Peer-review under responsibility of the Program Chairs.
doi:10.1016/j.procs.2014.05.535

parameters, is crucial for effective performance. e-Health is an example of M2M application where temporal constraints due to the real-time nature of critical medical applications is a stringent QoS requirement. So to ensure the respect of temporal properties of critical M2M applications, we need to put verification mechanisms.

In this paper, we aim to define a formal verification approach to validate temporal properties within Publish/ Subscribe M2M systems. To do so, we propose a formal graph-based model to characterize structural and timed properties of Publish/ Subscribe M2M systems on the top of the OM2M an interoperable M2M services platform. Based on the proposed model, we define a model checking based formal verification process of timed constraints.

The remainder of the paper is organized as follows: in section 2, a e-Health scenario is described as a business case. In section 3, some related works to our approach are introduced. In section 4, we emphasize the proposed approach for modeling and verifying temporal requirements of Publish/ Subscribe M2M systems. We consider the OM2M platform as the communication bus between M2M sensors and applications. Section5 develop a e-health scenario to validate our approach. Section 6 concludes and gives some perspectives as future works.

2. Use case: e-Health

In this work, we look at e-Health as a use case. The integrated use of telecommunications and information technology in the health sector leads to new challenges in transmitting and presenting health information in a timely and efficient manner for effective healthcare service performance. The quick delivery requirement of a patients measurements, a stringent requirement of medical application, is an important issue, especially in emergency situations.
In this paper, we consider the business case of monitoring the vital signs of patients with a long term chronic condition: heart failure. The objective of this scenario is the early recognition of heart attacks and fast treatment by combining vital parameters monitoring, emergency detection, and rescue management. Thereby, the system must ensure that the collection of data and the transmission of alarms respect some time related constraints. The patient is equipped with sensors which monitor physiological parameters such as heart rate, oxygen saturation and blood pressure. The healtcare team supervising the patient must receive alerts within hard temporal delays when the patient has exceeded the parameters set by the patient's practitioner. We notice that patient's vital signs measurements should be received by multiple doctors and caring personal with different temporal constraints. To illustrate these issues, we consider the example of a remote medical team supervising a group of patients. One patient's electrocardiogram shows heart tracing irregularities. Alarms should be transmitted to the emergency team within 400ms at most and the attending medical team with a maximum delay of 500ms.
Considering M2M systems where temporal constraints must be rigorously respected, our objective is to propose a formal approach for modelling and validating strict real-time M2M communications.

3. Related Works

In the literature, there is a number of works developed to support the management of Machine-to-Machine communications in particular time sensitivity of data delivery. [2] investigates on the management of M2M networks. Based on common characteristics of M2M applications, the authors identify the set of network and device management requirements of M2M applications taking into account activities from M2M standardization bodies. Considering time sensitivity of data, traffic classification and resource allocation based on the priority of M2M services are proposed as solutions. However, this work does not express a model for the management of timed requirements related to information exchanged between M2M machines.
To illustrate the criticality of timed related requirements of Machine-to-Machine communications, the e-Health domain is considered. In fact, e-Health services introduce stringent temporal constraints. Researchers conducted several works to enable timed data delivery. [3] proposes a mapping of QoS Class based on network Quality of Services (QoS) requirements for heterogenous eHealth services. The authors examine temporal properties for medical data transfer to specify classes for maximum delay considering various type of data service (audio, video, electrocardiogram, file transfer, etc.) for the healthcare services (tele-monitoring, tele-consultation, tele-diagnosis, etc). [4] introduces architecture and scenarios for QoS provisioning in emergency telemedicine. The authors pointed out the importance of data delivery of patients' vital signs in emergency situations where a delay will constitute a danger for the patient's life. They presents techniques and methods for IEEE 802.11 to ensure at time data delivery based on prioritizing

schema and bandwidth allocation. However, the authors rely only on the results of conducted experiments to verify non-violation of temporal requirements. They do not provide design-time techniques to validate system's temporal requirements. [5] propose an approach to the provision of QoS in an IP-based telemedicine system. To meet the latency requirements of real-time traffic, the autors rely on the DiffServ technology to minimize delay and jitter. They propose differentiated services for different classes of traffic by using mechanisms that include packet classification, policing, class-based queuing and scheduling, and random early detection. To validate the proposed solution, they rely on simulation's experiments.

4. Approach

In this paper, we are interested in the verification of Publish/ Subscribe M2M communications considering time related requirements. In this context, we first characterize Publish/ Subscribe communications on the top of the OM2M platform, which is an implementation of the ETSI M2M standard. Then, we introduce a formal graph based model to specify Publish/ Subscribe communications associated to time related requirements. Based on the defined formal model, we provide a model checking based techniques to validate time requirements for data exchange delays. In the following, we give an overview of our M2M based Publish/ Subscribe verification framework.

4.1. Global overview

The Figure1 illustrates the main activities of the proposed approach. It relies on two steps:
– Translation: that consists of the process of making the M2M architecture and temporal properties specification inputs in a format readable by the model checking tool. To model M2M architecture, we rely on graph grammar formalism and rewriting rules to transform a component based model into an event view to analyze the events driven in the system and their related temporal constraints. Then to process verification, the event based graph instance is translated into a timed automata where temporal requirements of data transmission are specified as CTL (Computation Tree Logic) formulas. To compute model transformation, we rely on the GMTE (Graph Matching and Transformation Engine)[6] and to enable model checking step, the model checker UPPAAL is used[7].
– Computation (Verification): that takes as inputs the specifications previously provided and produces an output "true" or "false" with a counter-example when an error is detected.

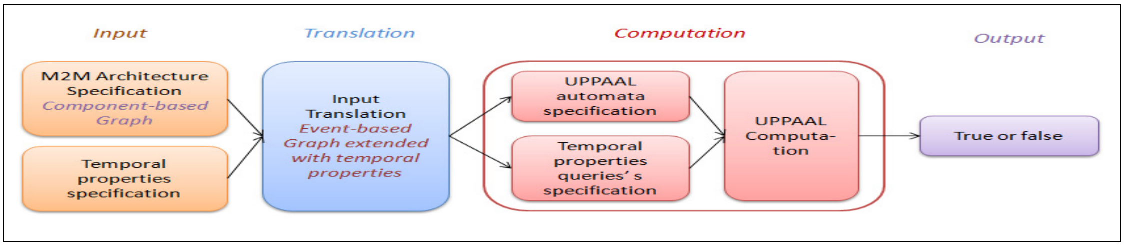

Fig. 1. A global view of the proposed approach

4.2. Graph grammar based model for Publish/ Subscribe communications

4.2.1. Publish/ Subscribe communications on the top of the OM2M platform

OM2M Architecture

To enable interoperable M2M communications between sensors and monitoring applications, OM2M platform [8] offers a standardized framework for developing services independently of the underlying network following the ETSI M2M standard [9]. OM2M platform satisfies most of M2M modeling requirements. However, this platform does not address the issue of timed QoS provisioning of M2M communications even though several M2M applications are evolving in hard real-time context. To overcome this limitation, we define a timed verification approach to enhance Publish/ Subscribe M2M communications reliability.

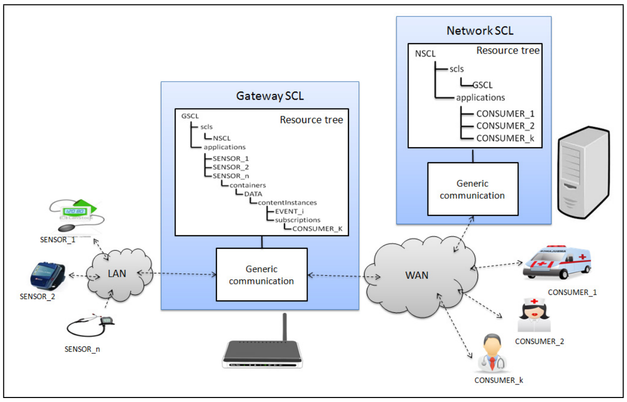

Fig. 2. A global view of the M2M architecture

As depicted in figure 2, the OM2M architecture is comprised of the following elements:
– Local Area Network (LAN): provides connectivity between devices (sensors) and M2M gateway.
– Wide Area Network (WAN): provides the access network to enable communications between M2M gateways and the M2M network.
– M2M Service capability layer (SCL): provides a horizontal layer offering M2M functions that are shared by different applications keeping them abstracted from their heterogeneity. SCs can be deployed on an M2M network, gateway or a device. In the above figure, we present the following service capabilities: *Generic Communication (GC)* (which represents the single point of contact for M2M communications which ensures secure application data delivery) and *Reachability, Addressing and Reporting (RAR)* (which provides persistence capability to store M2M resource states within the SCL and manage data required for M2M applications. The M2M resources, with their corresponding states, are organized on a resource tree structure.)
– M2M Gateway: acts as a proxy between M2M devices (that are connected to it) and the network domain.
– M2M network: provides M2M services and interconnection with M2M applications and M2M gateway.
– M2M application: is an application that runs the service logic and use M2M SCs accessible via an open interface. In our case study, monitoring applications of patients medical team are identified as M2M applications.
As ETSI M2M standard follows a Restful approach to handle data exchange, and given that a Restful architecture is about the transfer of representation of addressable resources, ETSI M2M standardized the resource structure. In fact, each SCL contains a resource structure where the information is registered. In our work, we are mainly interested in these ones:
– *application resource:* a software application that interacts with the M2M machines. It could represent a sensor or an actuator or a monitoring application.
– *container resource:*is used as a mediator for data buffering to enable data exchange between applications.
– *contentInstances resource:* the contentInstances resource represents a data instance in the container.
– *subscription resource:* stores information related to subscriptions for some resources. It describes a set of notifications a consumer is interested in and allows him to receive asynchronous notification when an event happens as the reception of new sensor event. In our case, subscribers are medical monitoring applications.

Publish/ Subscribe communication over OM2M platform

To ensure Publish/ Subscribe communications through the OM2M platform, we rely on the following subscription flow. Sensors and monitoring applications create their corresponding applications resources respectively on the M2M gateway and network resource trees. To publish sensed data, sensors' s applications create a DATA container and for each occurrence of an event, a new container's contentInstance resource is created. To receive events, consumers's applications initiate subscriptions resources by targeting contentInstances resources. Each time a new event is published, a notification is sent to all subscribed resources. The payload of the notification contains the representation of the published event. To summarize, sensors applications correspond to event producers, monitoring applications are event consumers and container' contentInstances resources are analogical to topics. As illustrated in figure 2, "SENSOR_i" and “CONSUMER_j” applications’ resources denote respectively sensors and medical team’ monitoring applications. The “CONSUMER_k” is subscribed to the DATA container’ contentInstances resource of “SENSOR_n”.

4.2.2. Temporal related requirements of Publish/ Subscribe communications

The objective of our approach is to validate Publish/ Subscribe M2M configurations taking into account temporal related requirements. We aim to design an M2M system where communications take place in required temporal delays. In the present work, we consider the following temporal properties:
– *Transmission Time*: It measures the time which a sensed data takes to be transported from the source to M2M gateway. We distinguish between Request Transmission Time (TT_{Req}) and Response Transmission Time (TT_{Resp}).
– *Execution Time (ET)*: It represents the time taken to perform processing which consists of managing M2M resources considering the received requests.
– *Response Time (RT)*: It measures the time between the service call and its acquittal. In this context, the response time represents the elapsed time between receipt of the request (creating a new contentInstance containing the sensed data) and when the response is returned acquitting the creation.
– *Service Time (ST)*: It measures the required time to prepare a notification to be sent to a set of subscribers.
– *Monitoring Period (MP_T)*: represents our temporal constraint. We want to check that the following temporal constraint: *a consumer requires to receive an event in a Period P of its issue. If the event is received after this period, it is no longer valid.* We note that a consumer can specify, according to its requirements, a different period for each topic on which he is subscribed.
Before introducing our approach, we give a description of graph based formalism.

4.2.3. Graph based formalism

To describe software architectures, graphs have the advantage of providing both intuitive and formally based formalism to describe a wide range of architectural structures [10]. In fact, graphs are widely used as informal means to forward the configuration of an application. In addition, their formal definition and the dissociation of the computation of entities and their coordination facilitate the checking of global properties of the system. A system configuration can be modelled using attributed graph, where vertices represent entities or events and edges specify their relationships. A graph is an instance of a graph grammar which specifies a configuration style. [11]

Definition 1: *Graph Grammar*
A graph grammar is defined as a system $< AX; NT; T; P >$ where:
– AX is the axiom;
– NT is a set of non-terminal vertices;
– T is a set of terminal vertices; and
– P is a set of graph grammar productions.
An instance of a graph grammar is obtained starting from the axiom AX by applying a sequence of productions P and where there is not non-terminal vertices. Graph-based models guarantee the correctness by construction of a software architecture by providing formal means for consistency’s preservation such as graph grammars and graph rewriting rules. You can see our previous work for more details about graph based formalism [12].

4.2.4. Proposed graph grammar for Publish/ Subscribe M2M communications

To validate the temporal requirements expressed above, the driven M2M events must be checked. A graph grammar is provided to generate an event-based configuration starting from a component-based description. The generated graph is labelled with the temporal properties.

In this paper, we study an M2M Publish / Subscribe architecture composed by N producers, M consumers communicating through the OM2M platform. For component-based model, we consider five types of component corresponding to *M2M gateway*, *M2M network*, *Producer*, *Consumer* and *Topic*. We consider that each node is identified by a unique identifier. These nodes are defined as non-terminal in the graph grammar. Two types of links between components are specified: 1) The first concerns the communication links that connect a consumer or a producer to the M2M gateway and network while 2) the second concerns the relationship of deployment, publication and subscription linking a topic respectively to the M2M gateway, producer and consumer. The communication edges are attributes with the following properties: Request Transmission Time and Response Transmission Time. As to the second category of links, we consider the temporal property: the Monitoring Period (associated to a required period specified by a consumer to receive an event notification).

The terminal nodes describe events. We consider that each node is identified by a unique identifier. We distinguish the following classes of events: "sending_evt", 'reception_evt", "reception_ack", "query_processing", "determination_subscribers" (the determination of the subscribers list to particular topic), "sending_notification" (the operation of sending notification to the set of consumers subscribed to a given topic once an event has been published on), "reception_notification". Each event node is characterized by a group of structural and temporal properties.

Two relationships classes between event nodes are distinguished: 1) The first concerns the temporal interval required to move from one event to another while 2) defines the temporal constraints between the transmission of an event and its reception. This class of links is characterized by a const attribute equal to "const_link".

We define the following graph grammar taking into account terminal and non terminal terms's characterization:
GG = (G, NT, T, P) where:
G *is the graph where nodes represent components.*
NT = *N(id_G, "M2MGateway", $idMachine_G$, ET_G, RT_G, ST_G), N(id_N, "M2MNetwork", $idMachine_N$, ET_N, RT_N, ST_N) ,N(id_P, "Producer", $idMachine_P$, ET_P, RT_P), N(id_C, "Consumer", $idMachine_C$, ET_C, RT_C), N(id_T, "Topic", TV), N("Temp")*, (The TV attribute denotes the topic value.)
T = *N(id_{se}, "sending_evt", id_P, TV), N(id_{re}, "reception_evt", id_P, TV), N(id_{ra}, "reception_ack", id_P, id_{se},TV), N(id_{qp}, "query_processing"), N(id_{ds}, "determination_subscribers", TV), N(id_{sn}, "sending_notification", TV), N(id_{rn}, "reception_notification", id_C, id_{se}, TV, MP_T)*, and
P = $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$

The specifications that have been elaborated for the relationships to be initiated or for a node to be instantiated, as well as how to do it, are described by the productions of the grammar. The production p_1 presents the initialization phase by converting a connection between the M2M gateway node and any topic deployed on it. We define the set of events corresponding to the required treatment when a date is received from a sensor. The production p_2 corresponds to the transformations related to the M2M gateway deploying any topic (different from that considered during initialization phase). This production is applied until no topic will be connected to the M2M gateway. The production p_3 represents the transformations related to a producer publishing on a topic. As mentioned previously, a producer can publish on different topics. So for a producer, we apply this rule as much he is connected to a topic.
The production p_4 and p_5 describe the transformations related to a consumer subscribed on a topic. A consumer specifies a different monitoring period for each topic on which it is subscribed. The production p_5 formalizes the suppression of a connection between a consumer and a topic.
The productions p_6 and p_7 specify the removal of "consumer" and "producer" nodes. These ones are deleted when they are no longer connected to a "topic" node.
The production p_8 expresses the removal of a "M2M gateway" and 'M2M network" nodes. These ones are deleted when they have no longer connections with "producer", "consumer" and "topic" node. These conditions are considered in NACs. The production p_9 is similar to the previous. It formalizes the deletion of a "topic" node. This one is eliminated when it has no more links with a "consumer", "producer" and "M2M gateway" nodes.
The production p_{10} describes the addition of the "const_link" edge. This link represents the time constraint valuated by the attribute "monitoring period". The production p_{11} is dedicated to the removal of the temporal nodes instantiated by the production p_5.

4.3. Temporal Verification using UPPAAL model checker

As our objective is is to validate temporal properties of M2M Publish/ Subscribe configurations by applying model checking techniques [13] using the UPPAAL model checker, we have to translate the event-based graph instance to a timed automata. To provide timed automata, we propose a transformation approach based on the following steps: definition of locations, guards, invariants, and final states.
In order to transform an event-based graph into a UPPAAL description that preserves the temporal constraints, we implemented an XML parser, which uses TinyXML C++ parser. Locations represent events defined in the event graph instance. Once the UPPAAL description is provided, we proceed to generating the CTL formulas corresponding to the temporal constraints. To do so, we will consider the event-based graph' s edges tagged with "const_link" attribute and

the value of the monitoring period. We aim to check if there exists a final state associated to a consumer in which the time required to receive an event has been less then the monitoring period specified by the consumer. The UPPAAL implementation of such temporal constraint is: $(E \langle \rangle \mathit{Process.final\_c1}\ \&\&\ t \langle 500)$ i.e: there is an execution leading to the consumer 1's final state where the monitoring period (the clock t) is lesser than 500u.t.

5. Illustrative scenario

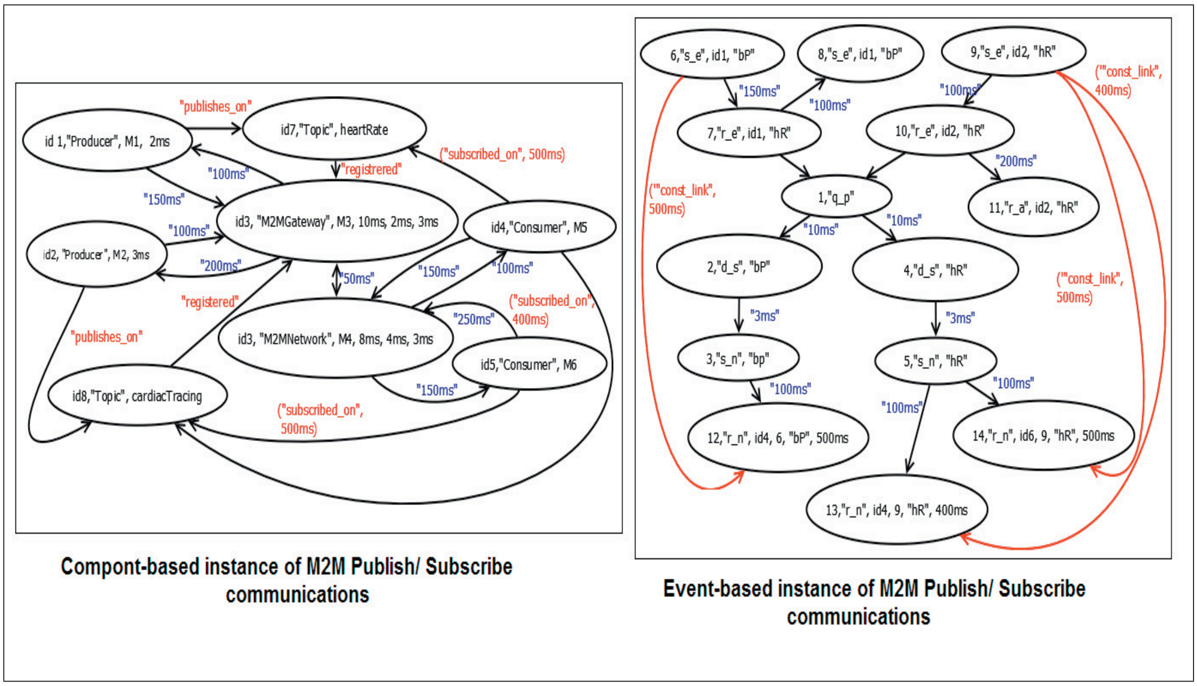

Fig. 3. Graph transformations of M2M Publish/ Subscribe communications considering temporal properties

This section describes an instantiation of our timed verification approach for Publish/ Subscribe M2M communications considering the e-health scenario described in section 2. The deployed Publish/ Subscribe M2M architecture is composed of two producers (the electrocardiogram and the heart rate monitor sensors) which publish patient vital signs' events on the topics (heart rate, and cardiac tracing) and two consumers representing the emergency service and the outpatient clinics'monitoring applications subscribed to these topics. Figure 3 depicts the component-based instance of the described Publish/ Subscribe architecture and the resulting event-based graph instance once the graph grammar's productions are applied. To check that the deployed architecture respect the specified hard temporal constraints, we proceed to the UPPAAL temporal properties checking. For lack of space, we give in figure 4 an extract of the generated timed automata where only the states of our M2M system following the receipt of an event are shown. Then, the UPPAAL implementations of data delivery's temporal constraints are checked. "False" or "True" output are supplied.

6. Conclusion

This paper points up the problem of temporal verification of Publish/ Subscribe M2M communications. To consider time related requirements of such systems, we propose a graph-based approach to model and characterize structural and temporal properties of components based M2M systems on the top of OM2M, a standardized platform for interoperable M2M communications. To conduct driven M2M events analysis, we rely on graph theory to transform a component-based view into an event-based one taking into account time related requirements associated with the

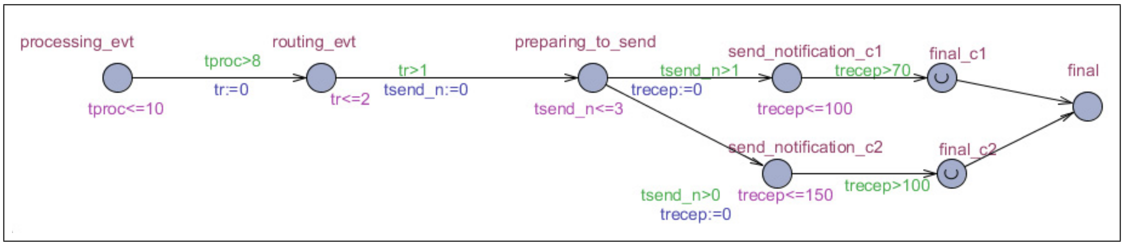

Fig. 4. An extract of the UPPAAL automata

occurrence of events. Then, we provide a model checking based techniques to validate time requirements for data exchange delays. The elaborated approach has been illustrated in e-Health use case to emphasize the criticality of temporal requirements.

As future work, we plan to enhance our work by dealing with the mobility support issue: the temporal related requirements should be assured by the different networks technologies. Furthermore, we plan to define a set of reconfiguration actions when temporal properties of M2M communication are not satisfied.

References

1. Boswarthick D, Elloumi O, Hersent O. M2M Communications: A Systems Approach, *Wiley*, 2012.
2. Pandey S, Mi-Jung C, Myung-Sup K, Hong J.W. Towards management of machine to machine networks, *Network Operations and Management Symposium (APNOMS)*, 2011 13th Asia-Pacific, pp.1,7, 21-23 Sept. 2011
3. Skorin-Kapov L, Matijasevic M. Analysis of QoS requirements for e-health services and mapping to evolved packet system QoS classes, *International Journal of Telemedicine and Applications*, 2010, p.1-18, January 2010
4. Zvikhachevskaya A, Markarian G, Mihaylova L, Quality of Service Consideration for the Wireless Telemedicine and E-Health Services, *Wireless Communications and Networking Conference*, 2009. WCNC 2009. IEEE, pp.1,6, 5-8 April 2009
5. Malindi P. QoS in Telemedicine, *Telemedicine Techniques and Applications, Prof. Georgi Graschew (Ed.)*, DOI: 10.5772/20240. Available from: http://www.intechopen.com/books/telemedicine-techniques-and-applications/qos-in-telemedicine
6. "GMTE: *Graph Matching and Transformation Engine"*, http://homepages.laas.fr/khalil/GMTE/
7. UPPAAL: http://uppaal.org/
8. "OM2M: *Open Machine-to-Machine Platform"*, http://om2m.org/
9. "*ETSI Functional Architecture Technical Report"*, http://www.etsi.org/deliver/etsi_ts/ 102600_102699/102690/01.01.01_60/ ts_102690v010101p.pdf
10. Le Metayer D. Describing software architecture styles using graph grammars, *IEEE Trans. Softw. Eng. 24* 1998, pp. 521-533.
11. Ehrig H. Tutorial introduction to the algebraic approach of graph grammars, *In Proceedings of the 3rd International Workshop on Graph-Grammars and Their Application to Computer Science*, pages 314, London, UK, 1987. Springer-Verlag.
12. Eichler C, Gharbi, G, Guermouche N, Monteil T, Stolf P. Graph-Based Formalism for Machine-to-Machine Self-Managed Communications, *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2013 IEEE 22nd International Workshop on , pp.74,79, 17-20 June 2013
13. Zhang P, Muccini H, Li B. A classification and comparison of model checking software architecture techniques, *Journal of Systems and Software*, Volume 83, Issue 5, May 2010, Pages 723-744, ISSN 0164-1212, http://dx.doi.org/10.1016/j.jss.2009.11.709.